
exosip2ctypes Documentation

Release 1.2.4

Liu Xue Yan

Apr 19, 2018, 6:09:45 AM

Contents

1	exosip2ctypes	3
1.1	exosip2ctypes package	3
2	Indices and tables	9
	Python Module Index	11

Build Time Apr 19, 2018, 6:09:44 AM

Version 1.2

Release 1.2.4

1.1 exosip2ctypes package

1.1.1 Submodules

<i>exosip2ctypes.call</i>	eXosip call API
<i>exosip2ctypes.context</i>	eXosip2 context API
<i>exosip2ctypes.error</i>	Error definitions
<i>exosip2ctypes.event</i>	eXosip2 event API
<i>exosip2ctypes.message</i>	
<i>exosip2ctypes.register</i>	eXosip2 REGISTER and Registration Management
<i>exosip2ctypes.sdp</i>	eXosip2 SDP helper API.
<i>exosip2ctypes.utils</i>	Some helper functions
<i>exosip2ctypes.version</i>	version information

exosip2ctypes.call

eXosip call API

This file provide the API needed to control calls. You can use it to:

- build initial invite.
- send initial invite.
- build request within the call.
- send request within the call.

This API can be used to build the following messages: INVITE, INFO, OPTIONS, REFER, UPDATE, NOTIFY

Classes

<code>Ack(context, did)</code>	default ACK for a 200ok received.
<code>Answer(context, tid, status)</code>	default Answer for request.
<code>InitInvite(context, to_url, from_url[, ...])</code>	default INVITE message for a create call.

exosip2ctypes.context

eXosip2 context API

Classes

<code>BaseContext</code>	
<code>Context([event_callback])</code>	Allocate and Initiate an eXosip context.
<code>ContextLock(context)</code>	A helper class for eXosip Context lock

exosip2ctypes.error

Error definitions

see `osip/include/osipparser2/osip_port.h`

Functions

<code>raise_if_osip_error(error_code[, message])</code>	raise an <code>OsipError</code> exception if <i>error_code</i> is not <code>OSIP_SUCCESS</code>
---	---

Exceptions

<code>MallocError</code>	Failed to allocate an <i>eXosip</i> context.
<code>OsipApiNotInitialized</code>	
<code>OsipBadParameter</code>	
<code>OsipDiskFull</code>	
<code>OsipError</code>	Base <i>Osip</i> error <code>Exception</code> Class
<code>OsipFileNotExists</code>	
<code>OsipNoCommonCodec</code>	
<code>OsipNoMem</code>	
<code>OsipNoNetwork</code>	
<code>OsipNoRights</code>	
<code>OsipNotFound</code>	
<code>OsipPortBusy</code>	
<code>OsipSyntaxError</code>	
<code>OsipTimeout</code>	
<code>OsipTooMuchCall</code>	
<code>OsipUndefinedError</code>	
<code>OsipUnknownError</code>	Osip error, but don't known the error code

Continued on next page

Table 1.5 – continued from previous page

OsipUnknownHost
OsipWrongFormat
OsipWrongState

exosip2ctypes.event

eXosip2 event API

see: http://www.antisip.com/doc/exosip2/group__eXosip2__event.html

Classes

Event(ptr, context)	Class for event description
EventType	Enumeration of event types

exosip2ctypes.message**Classes**

ExosipMessage(ptr, context)	class for eXosip2 message API
OsipMessage(ptr)	class for osip2 message API

exosip2ctypes.register

eXosip2 REGISTER and Registration Management

Classes

InitialRegister(context, from_, proxy[, ...])	initial REGISTER request.
Register(context, rid[, expires])	REGISTER request for an existing registration.

exosip2ctypes.sdp

eXosip2 SDP helper API.

Classes

SdpMessage(ptr)	SDP body
-----------------	----------

exosip2ctypes.utils

Some helper functions

Functions

<code>to_bytes(s[, encoding])</code>	Convert to <i>bytes</i> string.
<code>to_str(s[, encoding])</code>	Convert to <i>str</i> string.
<code>to_unicode(s[, encoding])</code>	Convert to <i>unicode</i> string.

Classes

<code>LoggerMixin</code>	Mixin Class provide a <code>logger</code> property
--------------------------	--

exosip2ctypes.version

version information

Functions

<code>get_library_version()</code>	return eXosip library (C library) version string
------------------------------------	---

1.1.2 Module contents

eXosip API

author Liu Xue Yan

email realtanbro@gmail.com

eXosip is a high layer library for rfc3261: the SIP protocol. It offers a simple API to make it easy to use. eXosip2 offers great flexibility for implementing SIP endpoint like:

- SIP User-Agents
- SIP Voicemail or IVR
- SIP B2BUA
- any SIP server acting as an endpoint (music server...)

If you need to implement proxy or complex SIP applications, you should consider using osip instead.

Here are the eXosip capabilities:

- REGISTER to `rpc_handler` registration.
- INVITE/BYE to start/stop VoIP sessions.
- INFO to send DTMF within a VoIP sessions.
- OPTIONS to simulate VoIP sessions.
- re-INVITE to modify VoIP sessions
- REFER/NOTIFY to transfer calls.
- MESSAGE to send Instant Message.

- SUBSCRIBE/NOTIFY to rpc_handler presence capabilities.
- any other request to rpc_handler what you want!

Constants

`exosip2ctypes.DLL_NAME`

Default so/dll name, value is `exosip2`

Functions

`exosip2ctypes.initialize` (*path: str=None*) → None:

Load *libeXosip2* into this Python library

Parameters `path` (*str*) – *libeXosip2* SO/DLL path, *default* is *None*. When *None* or empty string, the function will try to find and load so/dll by `DLL_NAME`

Raises `RuntimeError` – When failed loading so/dll

Attention: You **MUST** call this function **FIRST** to initialize *libeXosip2*, before any other actions!

CHAPTER 2

Indices and tables

- `genindex`
- `modindex`
- `search`

e

- `exosip2ctypes`, 6
- `exosip2ctypes.call`, 3
- `exosip2ctypes.context`, 4
- `exosip2ctypes.error`, 4
- `exosip2ctypes.event`, 5
- `exosip2ctypes.message`, 5
- `exosip2ctypes.register`, 5
- `exosip2ctypes.sdp`, 5
- `exosip2ctypes.utils`, 5
- `exosip2ctypes.version`, 6

D

DLL_NAME (in module exosip2ctypes), 7

E

exosip2ctypes (module), 6
exosip2ctypes.call (module), 3
exosip2ctypes.context (module), 4
exosip2ctypes.error (module), 4
exosip2ctypes.event (module), 5
exosip2ctypes.message (module), 5
exosip2ctypes.register (module), 5
exosip2ctypes.sdp (module), 5
exosip2ctypes.utils (module), 5
exosip2ctypes.version (module), 6

I

initialize() (in module exosip2ctypes), 7